CLUBE

Z80

Março/83

N.º 6

COORDENADORES: MARIA IRENE • ALBERTO FERNANDES • AV. BOAVISTA, 832 • TELEF. 65127 • 4100 PORTO

ÍNDICE

Edição: Clube Z80

Fotocomposição: Fotomecânica Mabreu/Porto

Impressão: Tip. Ramos dos Santos/Porto

Tiragem: 500 exemplares, Março 1983

PRIMEIRA PÁGINA

O QUE É O CLUBE Z-80

O CLUBE Z-80 é um grupo de pessoas possuidoras/utilizadoras de micro-computadores que usam e divulgam a sua aplicação.

O seu lançamento partiu de um pequeno número de pessoas que possuem o micro-computador ZX81 e que acharam do máximo interesse desenvolver a compatibilidade de programas provenientes de máquinas diferentes, mas com a mesma linguagem ao nível do microprocessador; Isto é especialmente importante para o utilizador do SINCLAIR ZX81 (e, principalmente a partir de agora, do ZX SPECTRUM) que é de longe a máquina mais expandida em Portugal.

A actividade do **CLUBE Z-80**, iniciada concretamente em Setembro/82, está ainda limitada à publicação desta revista mensal. Mas, pretende-se também:

- Desenvolver o interesse das pessoas nas áreas técnicas (telecomunicações, inteligência artificial, etc.).
- Dar aos associados a oportunidade de trocar experiências e comparar o uso de micro-computadores.
- Servir de suporte aos projectos mais interessantes de pequenas (ou grandes) alterações que possam surgir da parte dos utilizadores.
- Promover trocas de programas entre eles (quer descritivos, quer gravados).
- Lançar cursos de programação por correspondência, quer em Linguagem Basic, quer em Linguagem Máquina, para permitir um melhor aproveitamento da máquina utilizada (o curso em Basic já foi iniciado).
- Servir como suporte para esclarecimento e apresentar soluções para os problemas colocados pelos utilizadores das micro-máquinas, não só em termos de programa, mas também em termos de Hardware.

Associar os entusiastas das micro-máquinas, concretamente através do boletim **CLUBE Z-80** que todos recebem, não tem sido tarefa fácil. Apesar disso, o CLUBE reúne presentemente cerca de 140 pessoas, número de longe insuficiente para, pelo menos cobrir os custos que pressupõe — tanto mais quanto sabemos que em Portugal já foram vendidas cerca de 3000 unidades do micro-computador ZX81 e que as vendas do ZX SPECTRUM sobem dia após dia.

SE AINDA NÃO PERTENCE AO CLUBE Z-80, UTILIZE O CUPÃO DE INSCRIÇÃO INCLUÍDO NESTA REVISTA.

PASSO A PASSO — O "B-A-BA" DO BASIC

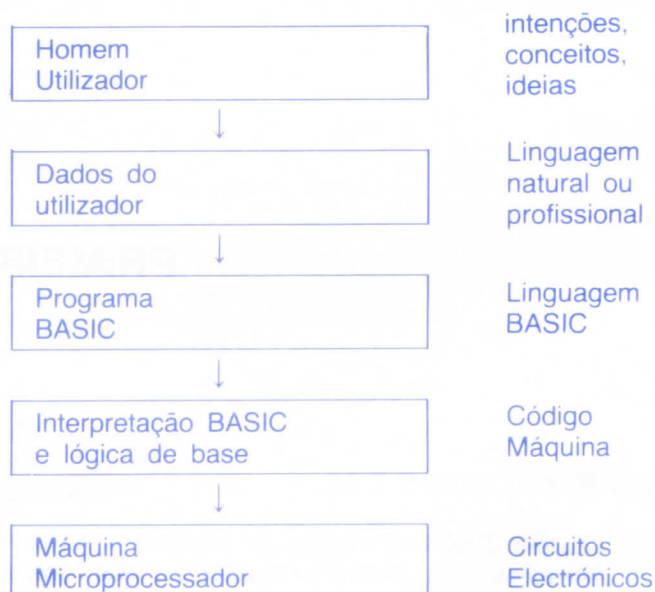
Comunicar com uma máquina é completamente diferente de um diálogo interpessoal. Antes de mais, é necessário fazer o esforço de adoptar uma linguagem que se considere igual à da máquina — por exemplo BASIC.
Efectivamente, BASIC não é a linguagem da máquina, mas um compromisso entre uma linguagem natural (no caso o Inglês) e o código interno da máquina. BASIC é uma linguagem simples: o essencial é saber usá-la correctamente. Thérèse Rieul expõe sete regras fundamentais a respeitar.

Todas as operações confiadas a um computador estão necessariamente a cargo do processador que contém, por um lado, a unidade aritmética e lógica, e por outro, a unidade de comando. O programa no interior da máquina exprime-se por uma sequência de valores, representando cada um um código interno reconhecido pela unidade de comando — só assim o computador sabe executar o programa. Qualquer programa escrito numa linguagem utilizada por comodidade de programação deve ser traduzido antes da execução.

Há dois tipos de métodos para a tradução:

1. Pode proceder-se a uma tradução global da totalidade do texto do programa escrito numa linguagem avançada (texto chamado texto origem ou programa origem) de modo a obter um programa executável, também chamado "programa objecto", "código objecto" ou "código máquina". Este programa pode ser guardado em disco, e posto na memória central para execução do computador. É o método adoptado frequentemente por linguagens como COBOL ("COmmon Business Oriented Language"), FORTRAN ("FORmula TRANslator"), PL/I ("Programming Language I") e muitas outras; estas Linguagens dizem-se compiladas, e os programas que as traduzem chamam-se compiladores.

2. Pode também proceder-se a uma tradução parcial, instrução por instrução, no momento da execução do programa: é pois o programa origem, eventualmente abreviado, que é armazenado no disco e se encontra contido na memória central para execução. Este método é adoptado por linguagens como APL ("A Programming Language") e BASIC na maioria dos computadores pessoais; estas linguagens são chamadas interpretadas, e os programas que as traduzem denominam-se interpretadores. Estes exemplos correspondem aos casos mais frequentes, pois é possível encontrar BASICs compilados, etc.

Na relação homem-máquina há um sistema hierárquico, que representamos simbolicamente assim:

O homem e a máquina estão respectivamente em cada uma das extremidades da cadeia. Todo os outros elos são as etapas de uma tradução, do mais geral ao mais específico, de uma formulação em linguagem natural até aos conteúdos binários da máquina.

Escrever um programa na linguagem BASIC é inserir-se num sistema, utilizando um tipo de comunicações muito mais acessível que o código interno da máquina. Esta vantagem implica naturalmente o respeito pelas regras de gramática e de vocabulário da lingugaem utilizada, não se podendc obter um programa executável, se elas não forem aplicadas.

Mas o mais importante é saber analisar o problema, organizar o seu programa e usar métodos que impeçam as "armadilhas".

A PRIMEIRA REGRA É DEFINIR O PROBLEMA ANTES DE LHE PROCURAR A SOLUÇÃO E A PROGRAMAR

Este enunciado parece tão absurdo, quanto evidente parece a regra; e portanto é ela que é mais frequentemente transgredida.

Nunca lhe aconteceu chegar ao teclado e escrever centenas de linhas em BASIC antes de enfrentar uma dificuldade importante que nunca tinha considerado e que, no entanto fazia parte da definição do problema?

Definir o problema não é o mesmo que escolher um método de resolução. É necessário, previamente, estabelecer de um modo preciso:

1. As informações que se querem obter e a sua forma (informações de saída).
2. Os dados de que se dispõe (dados de entrada).
3. As relações entre os dados de entrada e as informações de saída.

É nesta fase que convém pôr a questão fundamental: este problema releva de um tratamento informático ou teria uma solução manual simples?

Mesmo se, à partida, estiver convencido que a solução manual é de excluir, faça um esforço para descrever o tratamento tal como deveria ser feito manualmente, sem qualquer auxílio informático. Nessa altura talvez se aperceba que omitiu um ou vários elementos importantes de entre os dados necessários à entrada, ou entre as informações úteis de saída, ou simplesmente, que ignora certos aspectos do problema, de tal modo que não pode mesmo tratá-lo manualmente. Neste caso, consulte as fontes e recolha todas as informações.

Disponha de tempo para reflectir; oriente a sua reflexão não para o trabalho que vai redigir e os métodos a empregar, mas simplesmente para os resultados a atingir e os dados de entrada: estes dados que previu para entrada serão suficientes para obter os resultados previstos para saída (em todos os casos e em boas condições de utilização e segurança)?

A aplicação será cómoda para usar, inserindo-se sem dificuldades no trabalho quotidiano ou é necessário pensar em novos métodos de trabalho, em alterações de hábitos? Se sim, anote essa nova organização, não hesite em usar diagramas, verifique-a e passe depois à etapa seguinte.

Evite uma abordagem cega; esforce-se por considerar o problema no seu conjunto. Não caia na tentação de estudar imaturamente os detalhes excitantes de um dos aspectos do problema que conhece bem e que quer programar imediatamente.

Volte a sua atenção para aspectos do problema que o preocupem menos: é provavelmente neles que se encontram definições incompletas.

A SEGUNDA REGRA É ESTABELECER UM PLANO DO PROGRAMA ANTES DE COMEÇAR A ESCREVÊ-LO

Uma vez que o problema esteja perfeitamente definido, é necessário considerar várias abordagens para o resolver. Desde que consiga encontrar duas soluções, não se preocupe muito mais; ao fim de algum tempo de prática verá como fica surpreendido com a sua capacidade de encontrar alternativas realmente diferentes.

A solução deve ser independente dos dados e independente da linguagem de programação que vai ser utilizada, o que equivale a dizer que o tipo de solução depende da forma do problema e não do seu conteúdo. Uma tal solução denomina-se algoritmo*.

Num primeiro momento, convém deter-se nas grandes questões centrais e estabelecer uma estrutura de conjunto. Pode ser útil desenhar esta estrutura em forma de um esquema simbólico representando cada uma das funções ou grupos de operações e as suas relações entre elas. A este esquema chamamos organigrama — a representação simbólica de uma organização.

Convém conduzir o raciocínio seguindo a ordem lógica, partindo do geral para o particular. O algoritmo deve ser estabelecido por nível. Em cada nível, deve reter-se o que é mais simples e rejeitar, num nível inferior, o que for complexo e detalhado. Tem que se dividir as dificuldades em tantas partes quantas for possível, de modo a atenuar a complexidade e evitar dispersar inutilmente a atenção.

Numa primeira fase deve esforçar-se por elaborar um organigrama para cada nível. Se ele não ocupar uma folha dupla (29,7x42), é porque as dificuldades não foram completamente divididas. Nesse caso recomece — reaverá o tempo que julga ter perdido nessa fase. Logo que o seu método de resolução esteja bem determinado, reveja completamente tudo, de modo a certificar-se que não esqueceu nada.

Passemos agora à escrita do programa. Como vamos dispôr as diferentes partes do programa, e por que ordem se devem escrever as instruções?

A TERCEIRA REGRA É EVITAR UMA ABORDAGEM LINEAR NA ESCRITA DO PROGRAMA

A abordagem linear consiste em traduzir para linguagem BASIC as diferentes instruções na ordem cronológica da sua execução ulterior.

O programa que vai estabelecer não se destina apenas a ser executado pelo computador. Ele também será lido por si muitas vezes, pois o êxito não se obtém logo à primeira.

Dispõe de toda a memória viva do computador para incluir os diferentes elementos do seu programa, que se podem classificar em três grandes categorias:

- As inicializações, ou seja, as operações que só devem ser executadas no arranque do programa, tal como o formato da data.
- O programa principal e os diferentes tratamentos.
- Os sub-programas.

(continua no próximo número)

---

* O termo "algoritmo" provém da Idade Média de um matemático árabe ("Al Kow'rizmi") que estudou certos problemas para os quais não havia solução na sua época, descobrindo uma solução geral cuja forma dependia unicamente da forma do problema, independentemente dos valores específicos do enunciado do problema.

SECÇÃO DO LEITOR

DÚVIDAS. . .SUGESTÕES. . .COMENTÁRIOS. . .OPINIÕES. . .DÚVIDAS. . .SUGESTÕES. . .COMENTÁRIOS. . .

"Pretendo bibliografia (livros, artigos, etc.) sobre ROM para o ZX80. Se algum dos sócios a possuir, agradeço me comuniquem."

JOAQUIM CARRAPA
R. do Passadouro, 354 — Madalena
4405 Vila Nova de Gaia

"Como sugestão parece-me que a bibliografia indicada (no n.º anterior) poderia ter o preço (. . .)"

RUI LIMA
Setúbal

Relativamente à bibliografia que publicámos no boletim anterior, há alguns livros de que o CLUBE Z-80 ainda não dispõe. Passamos a especificar os preços dos livros que possuímos, e dos quais poderemos enviar fotocópias:

Manual de Basic ZX81	450$00
ZX81 Basic Book	350$00
The Explorers Guide to the Zx81	360$00
La Conduite du ZX81	290$00
Peek, Poke, Byte and Ram	250$00
The Gateway Guide to the ZX81 and ZX80	370$00
30 Programs For The ZX81	200$00
Getting Acquainted With Your ZX81	280$00
Sinclair ZX81 Rom Disassembly	280$00
The ZX81 Companion	280$00
Mastering Machine Code On Your ZX81	370$00
Machine Language Programming Made Simple For Your Sinclair	320$00
The ZX81 Pocket Book	280$00
What Can I Do With 1K-40 Programs & Routines	260$00
Hints & Tips For The ZX81	200$00
Understanding Your ZX81 Rom	350$00
The Sinclair ZX81 Programming For Real Aplications	400$00
Byteing Deeper Into Your ZX81	340$00
Not Only-30 Programs For The ZX81 . . .1K	300$00
Timex Sinclair 1000-User's Manual	300$00
50 Programas Educacionais	260$00
ZX Spectrum Introduction	200$00
ZX Spectrum Basic Programming	470$00
Manual do ZX Spectrum (em português)	470$00
The Spectrum Book Of Games	460$00
20 Best Programs For The ZX Spectrum	280$00
The ZX Spectrum Explored	450$00
The ZX Spectrum And How To Get The Most From It	375$00
60 Games and Applications For The ZX Spectrum	230$00
Over The Spectrum	400$00
Easy Programming For The ZX Spectrum	400$00
Programming Your ZX Spectrum	600$00

"Gostaria de obter as respostas a estas perguntas:

1. *Tenho um gravador PHILIPS stereo cuja entrada tanto para o microfone como para os auscultadores é feita através de uma ficha de 5 pinos. Será possível e proveitoso adaptar o referido gravador a um ZX81?*
2. *Ao trabalhar com o meu ZX81 sinto um problema nos ciclos FOR-NEXT: como hei-de armazenar por exemplo 30 variáveis denominadas A(1) . . . A(30) trabalhando fora do ciclo numa instrução apenas, com todas elas ao mesmo tempo?*
   Por exemplo —
   ```
   10 DIM A$(30)
   20 FOR B=1 TO 30
   30 LET A$(B) ="■"
   40 NEXT B
   ```
   Assim, uma linha 50 deveria ser PRINT . . . (todas as 30 variáveis A$).
3. *(. . .) Qual o número de bytes que me restam para programar se eu estiver em qualquer parte do programa, independentemente da memória que estiver a usar (1K, 16K, 64K)?*
4. *Alguns computadores têm a possibilidade de realizar desenhos de qualquer formato no écran. O Z×81 só tem a possibilidade de desenhar quadrados, meios quadrados, ¼ de quadrados, tanto pretos como constituídos (a sua área) por pontos. A minha pergunta é: poderei eu, através de Software, fazer os desenhos que quiser sem estar limitado aos caracteres do Z×81, ou seja, podendo eu fazer os meus próprios desenhos dos caracteres gráficos repartindo um caracter gráfico (por exemplo ■) nas partes que eu quiser?»*

RUI CARVALHO
Barreiro

1. Normalmente é desaconselhável o uso de gravadores stereo. No entanto, com o auxílio de um «conhecedor» dos pinos da ficha do gravador, poderá fazer a adaptação.
2. Não compreendemos completamente a sua pergunta. Se pretende imprimir 30 vezes o caracter «■», poderá usar as instruções:
   ```
   50 FOR B=1 TO 30
   60 PRINT «■»;
   70 NEXT B
   ```

Dará origem à impressão de uma linha contínua com o caracter em questão:

■ ■ ■ . . .etc.

Se retirar o ponto e vírgula da linha 60, terá 30 linhas com esse caracter.

3. Caso 1 — 1K de memória RAM — PRINT (17408 - (PEEK 16396+256*PEEK 16397-1))
   Caso 2 — 16K de memória RAM — PRINT (32768 - (PEEK 16396+256*PEEK 16397-1))
   Caso 3 — 64K de memória RAM — PRINT (65284 - (PEEK 16396+256*PEEK 16397-1))
4. A resposta é afirmativa: por software, pode fazer os traçados gráficos que pretender, em qualquer zona do écran definido pelo rectângulo 32×24 (colunas×linhas).
   Veja por exemplo o boletim do CLUBE n.º 0, pág. 4.

«Quais os Interfaces que o modelo B do microcomputador BBC possui?»

ANTÓNIO RAINHO
Porto

O modelo B está pronto para receber o interface que lhe permita usar uma impressora com entrada tipo paralelo «Centronics» ou RS423, mas não possui o interface.

O módulo para sintetização da voz e o de gráficos de alta resolução 640×256 também não estão englobados no modelo standard.

«(. . .) *Atendendo às carências de todos os utilizadores do Z80 de informação útil e em português (relativamente à linguagem máquina) para as suas experiências, tentarei da minha parte, ceder ao Clube muita da informação que disponho para ser publicada nas páginas do Boletim* (. . .)»

FERNANDO A. PRECES
Sacavém

O mini-curso de introdução à linguagem máquina da sua autoria é mais que bem-vindo! SERÁ PUBLICADO NO PRÓXIMO NÚMERO O PRIMEIRO CAPÍTULO.

Programa LABIRINTO

Autor: FERNANDO PRECES
Sacavém

```
   0■REM PROGRAMA TRAD.E MODIF.
POR ALMEIDA PRECES, EM 2/4/82.
   2 REM "5"
   5 GOSUB 1500
  10 LET U=0
  15 FAST
  20 FOR B=2 TO 28 STEP 2
  25 FOR A=0 TO 19
  30 PRINT AT A,B;"■"
  40 NEXT A
  45 PRINT AT RND*14+3,B;" "
  50 NEXT B
  60 FOR A=0 TO 30
  70 PRINT AT 0,A;"■";AT 19,A;"■
"
  80 NEXT A
  90 FOR B=1 TO 18
 100 PRINT AT B,0;"■";AT B,30;"■
"
 110 NEXT B
 120 GOSUB 1000
 125 GOTO 167
 150 PRINT AT A,B;"■"
 155 LET Z=Z-673
 160 PRINT AT 20,0;"PONTUACAO: "
;Z;"     "
 165 RETURN
 167 LET Z$=INKEY$
 170 IF Z$="" THEN LET Z$=A$
 180 LET Z=Z-50
 200 LET Y=A
 210 LET X=B
 217 LET R=RND*RND*RND
 220 LET A=A+(Z$="Z")-(Z$="Q")
 230 LET B=B+(Z$="L")
 232 PRINT AT Y,X;" "
 233 IF PEEK (PEEK 16396+256*PEE
K 16397+33*A+B+1)=128 THEN GOSUB
 150
 235 PRINT AT A,B;"*"
 255 IF A>18 OR A<2 OR B<1 THEN
LET Z=INT (Z/3)
 260 IF A>18 OR A<2 OR B<1 OR B>
29 THEN GOTO 510

 490 LET A$=Z$
 500 GOTO 167
 510 PRINT AT 20,0;"FIM DO RONDE
 PONTOS: ";INT (10000/Z)
 520 IF Z>U THEN LET U=Z
 530 FOR Q=1 TO 6
 540 PRINT AT 21,3;"PENALIZACAO:
";U
 545 PRINT AT A,B;"■";AT A,B;"■"
AT A,B;"■"
 550 PRINT AT 21,3;"A MENOR PENA
LIZACAO: ";U
 560 FOR O=0 TO 400
 565 NEXT O
 570 CLS
 580 PRINT AT 17,0;"QUER TENTAR
OUTRA VEZ?(DIGA S/N)"
 585 INPUT O$
 590 CLS
 600 IF O$="S" THEN RUN 10
 610 PRINT AT 17,0;"NAO QUER ? E
NTAO ATE A PROXIMA."
 620 STOP
1000 LET A=10
1010 LET B=1
1020 LET Z=20000
1030 LET Y=A
1040 LET X=B
1050 LET A$="Z"
1100 SLOW
1200 RETURN
1500 PRINT AT 2,8;"LABIRINTO"
1505 PRINT ,,"NESTE JOGO, SAO OS
 SEUS REFLEXOS"
1510 PRINT "E  A PRECISAO DOS SE
US  DEDOS NO"
1515 PRINT "TECLADO, QUE SERAO P
ONTUADOS."
1520 PRINT ,,"UM ROBOT E POR SI
COMANDADO PARA"
1525 PRINT "SE DESLOCAR PELOS CO
RREDORES SEM"
1530 PRINT "TOCAR NAS PAREDES."
1535 PRINT ,,"A TECLA ■ MOVE PAR
A CIMA, A ■ PA"
1540 PRINT "RA BAIXO, A ■ PARA O
 LADO  E A ■"
1545 PRINT "TRAVA O MOVIMENTO."
1550 PRINT ,," PRIMA ""N/L"", PA
RA COMECAR."
1585 INPUT O$
1590 CLS
1595 RETURN
```

Programa SISTEMA DE CONVERSÃO DE BASES DEC-HEX-DEC

Autor: FERNANDO PRECES
Sacavém

```
   1 REM 0123456789ABCDEF
   5 REM ELABORADO POR FERNANDO
PRECES (NOV.1982)
  10 GOSUB 400
  15 IF A=2 THEN GOTO 85
  18 REM DECIMAL-HEXADECIMAL
  20 DIM X(6)
  25 LET T=24
  28 INPUT DC
  29 PRINT AT 13,5;DC;TAB 11;" =
"
  30 LET K=DC/16
  40 FOR N=1 TO 6
  50 GOSUB 500
  60 NEXT N
  70 SCROLL
  75 SCROLL
  80 GOTO 25
  85 REM HEXADECIMAL-DECIMAL
  90 DIM H(6)
  95 GOSUB 200
  98 LET R=0
 100 INPUT A$
 110 PRINT AT 13,5;A$;TAB 11;" =
"
 120 FOR N=1 TO (LEN A$)
 125 LET M=(LEN A$)-(N-1)
 130 LET H(N)=VAL A$(M TO M)
 140 LET P=H(N)*16**(N-1)
 150 LET R=R+P
 160 NEXT N
 170 PRINT AT 13,16;R
 175 SCROLL
 180 SCROLL
 190 GOTO 98
 210 LET A=10
 220 LET B=A+1
 230 LET C=B+1
 240 LET D=C+1
 250 LET E=D+1
 260 LET F=E+1
 270 RETURN
 400 PRINT AT 2,3;"CONVERSAO (BA
SES NUMERICAS)."
 405 PRINT AT 5,3;"1 - DECIMAL-H
EXADECIMAL"
 410 PRINT AT 7,3;"2 - HEXADECIM
AL-DECIMAL"
 415 PRINT AT 10,2;"INTRODUZA O
NUMERO DESEJADO."
 420 PRINT AT 13,2;"EM SEGUIDA I
NTRODUZA O NUMERO"
 425 PRINT AT 15,2;"A CONVERTER.
"
 428 PRINT AT 17,2;"(MAXIMO ATE
6 ALGARISMOS)."
 430 INPUT A
 435 CLS
 440 RETURN
 500 LET X(N)=(K-INT K)*16
 510 LET K=INT K/16
 520 GOSUB 600
 530 RETURN
 600 LET R=PEEK (16514+X(N))
 605 PRINT AT 13,T;CHR$ R
 610 LET T=T-2
 630 RETURN
```

Programa SISTEMAS DE EQUAÇÕES CÚBICAS

Autor: FERNANDO PRECES
Sacavém

```
   0 REM PROGRAMA ELABORADO POR
ALMEIDA PRECES EM 6/6/1982
   2 REM "3"
   5 PRINT ,,"SISTEMAS DE EQUACO
ES CUBICAS"
  10 PRINT ,,,,"1 - EQUACAO CUBI
CA"
  15 PRINT ,,"2 - RAIZ DUMA FUNC
AO (PELO METO-     DO DE NEWTON)"
  20 PRINT ,,"3 - IDEM (METODO D
O PONTO MEDIO)"
  30 INPUT S
  35 CLS
  40 IF S=1 THEN GOTO 100
  45 IF S=2 THEN GOTO 400
  50 IF S=3 THEN GOTO 600
 100 GOSUB 3000
 110 PRINT ,,,,"█ A(X**3)+B(X**2
)+CX+D=0 █"
 112 PRINT AT 15,8;"PRIMA N/L."
 113 INPUT B$
 114 CLS
 115 GOSUB 2000
 155 LET P=C/(3*A)-B*B/(9*A*A)
 160 LET Q=2*B*B*B/(27*A*A*A)-B*
C/(3*A*A)+D/A
 165 LET CC=Q*Q+4*P*P*P
 170 IF CC>=0 THEN GOTO 180
 175 GOTO 310
 180 LET H=(Q+SQR CC)/2
 185 IF H>=0 THEN GOTO 200
 190 LET R=(-H)**(1/3)
 193 LET R=-R
 195 GOTO 210
 200 LET R=H**(1/3)
 210 LET H=(Q-SQR CC)/2
 215 IF H>=0 THEN GOTO 235
 220 LET I=(-H)**(1/3)
 225 LET I=-I
 230 GOTO 245
 235 LET I=H**(1/3)
 245 IF CC=0 THEN GOTO 255
 250 GOTO 280
 255 PRINT ,,"█ SOLUCAO: █"
 258 PRINT ,,"** 2 RAIZES REAIS:
**"
 260 LET J=R+I
 265 PRINT ,," Y(1) = ";-J-B/(3*A
)
 270 PRINT ,," Y(2) = ";J/2-B/(3*
A)
 272 INPUT B$
 276 CLS
 278 GOTO 5
 280 LET J=(R+I)
 282 PRINT ,,"█ SOLUCAO: █"
 283 PRINT "** 1 RAIZ REAL E 2 I
MAGINARIAS**"
 285 PRINT ,,"▪ RAIZ REAL: ▪"
 290 PRINT ,," = ";-J-(B/(3*A))
 296 PRINT ,,"▪ PARTE REAL: ▪"
 298 PRINT ,," = ";J/2-(B/(3*A))
 300 PRINT ,,"▪ PARTE IMAGINARIA
▪"
 302 PRINT ,," = ";(SQR 3*(R-I))
/2
 304 INPUT B$
 306 CLS
 308 GOTO 5
 310 LET J=SQR -(P)
 315 LET M=Q/(2*J*J*J)
 320 LET V=((ACS M)*180/PI)/3
 325 LET I=0
 326 PRINT ,,"█ SOLUCAO: █"
 328 PRINT ,,"█ 3 RAIZES REAIS:
█"
 330 FOR T=1 TO 3
 340 LET I=120+I
 345 LET N=(COS ((V+I)/180*PI))*
(-2)*J-B/3*A
 350 PRINT ,,"▪ X(";T;")= ";N
 355 NEXT T
 360 INPUT B$
 370 CLS
```

```
375 GOTO 5
400 PRINT ,," METODO DE NEUTON"
405 GOSUB 3000
410 PRINT ,," F(X)=A(X**3)+B(X*
*2)+CX+D"
415 INPUT B$
420 CLS
425 GOSUB 2000
430 INPUT EO
435 PRINT "**ENTRADA 5**█EO= ";
EO
440 INPUT X
445 PRINT "**ENTRADA 6**█XO= ";
X
450 INPUT H
455 PRINT "**ENTRADA 7** █H= ";
H
460 LET XO=X
465 LET J=2
470 LET F=A*X*X*X+B*X*X+C*X+D
475 LET X=X+H
480 LET J=J-1
485 IF J=0 THEN GOTO 500
490 LET G=F
495 GOTO 470
500 LET F=(F-G)/H
505 LET G=XO-G/F
510 LET I=ABS (G-XO)
515 IF I>=EO THEN GOTO 525
520 GOTO 535
525 LET X=G
530 GOTO 460
535 PRINT ,,"█ SOLUCAO: █"
540 PRINT ,," X= ";XO
545 INPUT B$
550 CLS
555 GOTO 5
600 PRINT ,," METODO DO PONTO M
EDIO"
605 GOSUB 3000
615 PRINT ,," F(X)=X**3+X**2-X-
1"
620 INPUT B$
625 CLS
630 INPUT A
635 PRINT "**ENTRADA 1** █A= ";
A
640 INPUT B
645 PRINT "**ENTRADA 2** █B= ";
B
650 INPUT E
655 PRINT "**ENTRADA 3** █E= ";
E
660 LET C=ABS (B-A)
665 IF C>=E THEN GOTO 675
670 GOTO 715
675 LET D=(A+B)/2
680 GOSUB 4000
685 IF F>=0 THEN GOTO 705
690 LET A=D
700 GOTO 660
705 LET B=D
710 GOTO 660
715 PRINT ,," █ SOLUCAO. █"
720 PRINT ,," X= ";D
725 INPUT B$
730 CLS
735 GOTO 5
2000 PRINT "**ENTRADA ""A""= ";
2005 INPUT A
2010 PRINT A
2015 PRINT ,,"**ENTRADA ""B""= "
;
2020 INPUT B
2025 PRINT B
2030 PRINT ,,"**ENTRADA ""C""= "
;
2035 INPUT C
2040 PRINT C
2045 PRINT ,,"**ENTRADA ""D""= "
;
2050 INPUT D
2055 PRINT D
2080 RETURN
3000 PRINT ,," RESOLUCAO DUMA EQ
UACAO CUBICA"
3010 PRINT ,," PELA FORMULA:"
3019 RETURN
4000 LET F=D*D*D+D*D-D-1
4010 RETURN
```

Programa HORÁRIO

Autor: Manuel Quinaz
Porto

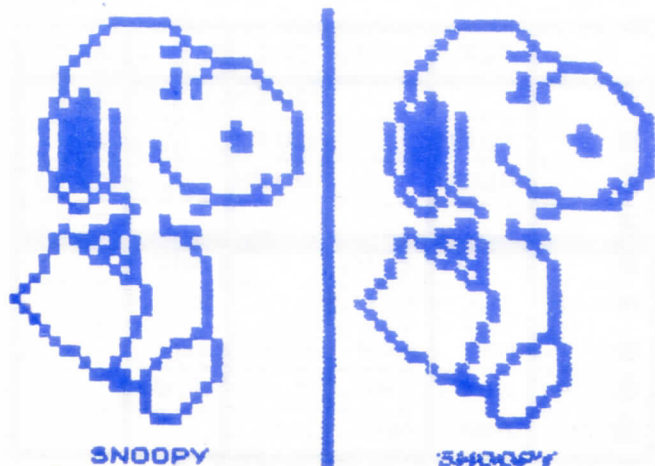
```
  5 REM "HORARIO"
 10 FOR I=0 TO 31
 20 PRINT AT 0,I;"█"
 30 PRINT AT 20,I;"█"
 40 NEXT I
 50 FOR I=0 TO 20
 60 PRINT AT I,0;"█"
 70 PRINT AT I,31;"█"
 80 NEXT I
 90 PRINT AT 0,0;"█";AT 0,31;"█
";AT 20,0;"█";AT 20,31;"█"
100 FOR I=1 TO 30
110 PRINT AT 2,I;"█"
120 PRINT AT 4,I;"-";AT 6,I;"-"
;AT 8,I;"-";AT 10,I;"-";AT 12,I;
"-";AT 14,I;"-";AT 16,I;"-";AT 1
8,I;"-"
130 NEXT I
140 PRINT AT 2,0;"█";AT 2,31;"█
"
150 FOR I=1 TO 20
160 PRINT AT I,6;"█";AT I,11;"█
";AT I,16;"█";AT I,21;"█";AT I,2
6;"█"
170 NEXT I
180 PRINT AT 0,6;"█";AT 0,11;"█
";AT 0,16;"█";AT 0,21;"█";AT 0,2
6;"█";AT 20,6;"█";AT 20,11;"█";A
T 20,16;"█";AT 20,21;"█";AT 20,2
6;"█"
190 PRINT AT 2,6;"█";AT 2,11;"█
";AT 2,16;"█";AT 2,21;"█";AT 2,2
6;"█"
200 LET A$="█ SEG.█TER.█QUA.█QU
I.█SEX.█SAB."
210 PRINT AT 1,0;A$
220 PRINT AT 3,1;"█";AT 5,1;"█"
;AT 7,1;"█";AT 9,1;"█";AT 11,1;"
█";AT 13,1;"█";AT 15,1;"█";AT 17
,1;"█";AT 19,1;"█"
230 LET B$="████████████ALMOCO██
████████████"
240 PRINT AT 10,0;B$
250 LET Q$="     █F.Q.█E.V.█
█MAT."
260 PRINT AT 11,2;Q$
270 LET W$="FRA.█ING.█E.V.█BIO.
█BIO."
280 LET E$="MAT.█MAT.█POR.█POR.
█ING."
290 LET R$="POR.█GEO.█FRA.█F.Q.
█F.Q."
300 LET T$="BIO.█FRA.█MAT.█HIS.
"
310 PRINT AT 13,2;W$;AT 15,2;E$
;AT 17,2;R$;AT 19,2;T$
320 LET Y$="     █     █     █QUIM
█    █HIS"
330 LET U$="     █QUIM█     █QUIM
█    █GEO."
340 LET I$="     █QUIM█     █QUIM
█    █E.F."
350 PRINT AT 3,2;Y$;AT 5,2;U$;A
T 7,2;I$;AT 9,27;"E.F."
360 PRINT AT 21,0;"HORARIO DO 9
ANO TURMA G"
```

	SEG.	TER.	QUA.	QUI.	SEX.	SAB.
1				QUIM		HIS
2		QUIM		QUIM		GEO.
3		QUIM		QUIM		E.F.
4						E.F.
ALMOCO						
1		F.Q.	E.V.		MAT.	
2	FRA.	ING.	E.V.	BIO.	BIO.	
3	MAT.	MAT.	POR.	POR.	ING.	
4	POR.	GEO.	FRA.	F.Q.	F.Q.	
5	BIO.	FRA.	MAT.	HIS.		

HORARIO DO 9 ANO TURMA G

Programa SNOOPY

Autor: Manuel Quinaz
Porto

```
5 REM "SNOOPY"
7 FOR I=0 TO 17 STEP 17
10 PRINT AT 0,I;"
20 PRINT AT 1,I;"
30 PRINT AT 2,I;"
40 PRINT AT 3,I;"
50 PRINT AT 4,I;"
60 PRINT AT 5,I;"
70 PRINT AT 6,I;"
80 PRINT AT 7,I;"
90 PRINT AT 8,I;"
100 PRINT AT 9,I;"
110 PRINT AT 10,I;"
120 PRINT AT 11,I;"
130 PRINT AT 12,I;"
140 PRINT AT 13,I;"
150 PRINT AT 14,I;"
160 PRINT AT 15,I;"
170 PRINT AT 16,I;"
180 PRINT AT 17,I;"
190 PRINT AT 18,I;"
200 PRINT AT 19,I;"
210 PRINT AT 21,I+4;"SNOOPY"
220 NEXT I
230 FOR I=0 TO 21
240 PRINT AT I,15;"█"
250 NEXT I
```

GRÁFICOS NO SPECTRUM

A SINCLAIR forneceu-nos um vantajoso comando de desenho gráfico para o SPECTRUM, desprezando no entanto desenhos de quadrados, triângulos, pentágonos e outros, devido à simplicidade com que é possível obtê-los.

Na lista que se segue, pode ver as variáveis necessárias que se estabelecem antes da chamada sub-rotina. São:

SIDES — Número de lados da figura
Ex.: LET SIDES=5, o que deve dar um pentágono.

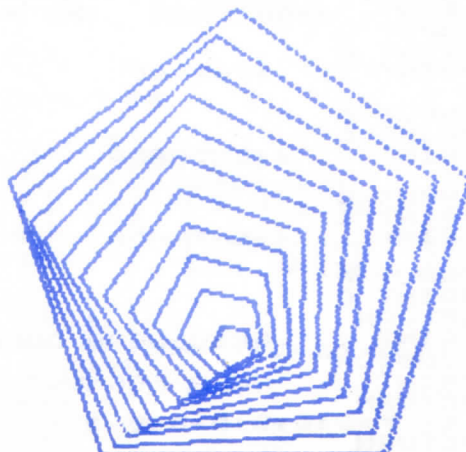
LENGTH — Comprimento de cada lado da figura
Ex.: LET LENGTH=20, dá lados de comprimento 20.

ANGLE — Ângulo entre o primeiro lado e a linha horizontal da figura (em radianos)
Ex.: LET ANGLE=$\pi/6$ dá uma inclinação de 30°.

Cada uma destas variáveis é estabelecida por GOSUB 1000 que desenha a figura na posição do cursor.

Nestas três demonstrações (que apresentamos a seguir), o primeiro programa apresenta um pentágono com uma aproximação tri-dimensional. O segundo, «HONEYCOMB», preenche o écran com hexágonos unidos formando «favos de mel». O terceiro mostra como um polígono pode gerar outros polígonos em desenhos, todos eles interessantes.

Tente substituir as instruções «DATA» com outros valores da sua imaginação, produzindo assim as suas próprias figuras.

```
   5 BORDER 0: PAPER 0: INK 6: CLS : LET sides=5
  10 FOR h=0 TO 100 STEP 10
  20 PLOT h/2+60,h*COS PI/3: LET length=110-h: LET angle=h/200
  30 GO SUB 1000
  40 NEXT h
  50 STOP
 999 REM GRAPHICS SUBROUTINE
1000 FOR S=0 TO 2*PI-.1 STEP 2*PI/sides
1010 DRAW length*COS (angle+s),length*SIN (angle+s)
1020 NEXT s
1030 RETURN
```

```
   1 REM HONEYCOMB
  10 BORDER 0: PAPER 0: INK 6: C
LS : LET sides=6: LET length=10:
 LET angle=0
  20 FOR h=10 TO 230 STEP 32
  30 FOR g=10 TO 150 STEP 18
  40 PLOT h,g: GO SUB 1000
  50 PLOT h+16,g+9: GO SUB 1000
  60 NEXT g: NEXT h
 900 STOP
 999 REM GRAPHICS SUBROUTINE
1000 FOR S=0 TO 2*PI-.1 STEP 2*P
I/sides
1010 DRAW length*COS (angle+s),l
ength*SIN (angle+s)
1020 NEXT s
1030 RETURN
```

```
   5 BORDER 0: PAPER 0: CLS
  10 FOR n=1 TO 5: LET angle=0:
READ ink,sides,step,length
  20 INK ink: CLS
  30 PLOT 126,87: GO SUB 1000
  40 LET angle=angle+step
  50 IF angle<2*PI THEN GO TO 30
  60 PAUSE 150: NEXT n
  90 RESTORE 3000
 100 FOR n=1 TO 5: LET angle=0:
READ ink,sides,step,length,centr
e
 120 INK ink: CLS
 130 PLOT 126+centre* COS  angle
,87+centre*SIN angle: GO SUB 100
0
 140 LET angle=angle+step
 150 IF angle<2*PI THEN GO TO 13
0
 160 PAUSE 150: NEXT n
 170 RUN
 999 REM GRAPHICS SUBROUTINE
1000 FOR S=0 TO 2*PI-.1 STEP 2*P
I/sides
1010 DRAW length*COS (angle+s),l
ength*SIN (angle+s)
1020 NEXT s
1030 RETURN
1998 REM DATA FOR SHAPES
1999 REM COLOURS AND SIZES.
2000 DATA 7,5,2*PI/5,50
2010 DATA 2,6,PI/3,40
2020 DATA 4,6,PI/6,40
2030 DATA 6,5,PI/5,50
2040 DATA 5,7,PI/7,40
2050 DATA 3,4,PI/8,60
2998 REM SAME DATA BUT
2999 REM CENTRE MOVES ABOUT.
3000 DATA 7,5,2*PI/5,50,-25
```

```
3010 DATA 2,6,PI/3,40,-20
3020 DATA 4,6,PI/6,40,-20
3030 DATA 6,5,PI/5,50,-25
3040 DATA 5,7,PI/7,40,-20
3050 DATA 3,4,PI/8,70,-35
```

OS COMANDOS USADOS COM GRÁFICOS SPECTRUM

BORDER — Representa a côr da área que circunda o rectângulo principal.

PAPER — Trata a área principal (rectângulo central) o écran.

INK — Côr que se pretende usar.

0 — Negro
1 — Azul
2 — Vermelho
3 — Magenta (púrpura)
4 — Verde
5 — Cyan (azul-verde)
6 — Amarelo
7 — Branco

Exemplo:

PRINT INK 2; PAPER 1; «LOG»

Aparecerá no seu SPECTRUM a palavra LOG escrita em vermelho, sobre fundo azul.

Podemos agora combinar

BORDER n...

PAPER m...

INK p...

de diferentes formas para obtermos os diversos efeitos destes comandos.

DRAW — Este comando permite-lhe traçar uma linha desde o ponto actual de exibição do écran até ao ponto que dista daquele **X** posições horizontais, **Y** verticais, com um ângulo de deslocamento **Z**.

Programas, CONE e CILINDRO

Estes pequenos programas mostram como representar CONES e CILINDROS no **SPECTRUM**, e assim dar a ideia de «sombra», para melhor representar um sólido geométrico.

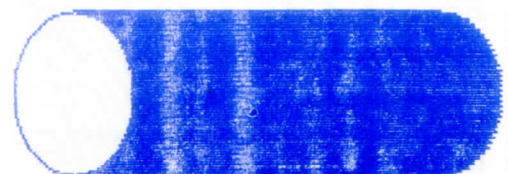
Para desenhar o cilindro, planificou-se o mesmo com papel quadriculado, marcando as posições para PLOT e PRINT. O comprimento dos eixos maior e menor pode ser facilmente alterado.

As linhas 60 a 140 e 400 a 510 desenham as elipses, enquanto as linhas 795 a 812 desenham uma série de traços com diferentes comprimentos.

Programa CONE

```
   1 REM "cone"
   2 PAPER 7: INK 0: BORDER 2: C
LS
  60 DEF FN v(n,m,a)=INT SQR ((m
↑2)-((m↑2/n↑2)*(a↑2)))
  70 DEF FN w(n,m,a)=INT SQR ((n
↑2)-((n↑2/m↑2)*(a↑2)))
  80 LET m=32: LET n=16: LET r=1
27: LET s=47
  90 FOR a=0 TO 20 STEP 1
 100 LET y=FN v(m,n,a)
 110 GO SUB 400
 240 FOR a=12 TO 0 STEP -.5
 250 LET x=FN w(m,n,a)
 270 GO SUB 500
 350 GO TO 620
 400 PLOT r-a,s-y: PLOT a+r,s+y
 410 PLOT r-a,s+y: PLOT a+r,s-y
 420 NEXT a: RETURN
 500 PLOT r-x,s-a: PLOT r+x,s+a
 510 PLOT r-x,s+a: PLOT r+x,s-a
 520 NEXT a
 530 RETURN
 620 FOR a=4 TO 30 STEP 4
 630 LET y=FN v(m,n,a)
 635 PLOT r-a,s+y: DRAW 127+a-r,
135-s-y
 637 NEXT a
 650 FOR a=1 TO 19 STEP 6
 651 LET y=FN v(m,n,a)
 652 PLOT r+a,s+y: DRAW 127-a-r,
135-s-y
 654 NEXT a
 685 PLOT 95,47: DRAW 32,88
 690 PLOT 159,47: DRAW -32,88
```

Programa CILINDRO

```
   1 REM "cylinder"
   2 PAPER 4: INK 6: BORDER 0: C
LS
  60 DEF FN v(n,m,a)=INT SQR ((m
↑2)-((m↑2/n↑2)*(a↑2)))
  70 DEF FN w(n,m,a)=INT SQR ((n
↑2)-((n↑2/m↑2)*(a↑2)))
  80 LET m=24: LET n=32: LET r=5
5: LET s=87
  90 FOR a=0 TO 17 STEP 1
 100 LET y=FN v(m,n,a)
 110 GO SUB 400
 130 FOR a=22 TO 0 STEP -1
 140 LET x=FN w(m,n,a): GO SUB 5
00
 150 GO TO 795
 400 PLOT r-a,s-y: PLOT a+r,s+y
 410 PLOT r-a,s+y: PLOT a+r,s-y
 420 NEXT a: RETURN
 500 PLOT r-x,s-a: PLOT r+x,s+a
 510 PLOT r-x,s+a: PLOT r+x,s-a
 520 NEXT a: RETURN
 795 FOR a=0 TO 15 STEP 1
 796 LET y=FN v(m,n,a)
 801 PLOT a+r,s+y: DRAW 144,0
 803 PLOT a+r,s-y: DRAW 144,0
 804 NEXT a
 806 FOR a=23 TO 0 STEP -1
 807 LET x=FN w(m,n,a)
 808 PLOT r+x,s+a: DRAW 144,0
 811 PLOT r+x,s-a: DRAW 144,0
 812 NEXT a
```

Programa ANALISADOR DE ÁREAS (ZX-SPECTRUM)

Autor: SANTIAGO RIBAS
Porto

Este programa pode servir comc comparador de vários projectos de um mesmo tipo de edifício já que ele apresenta no final a listagem das áreas das várias funções do edifício em metros quadrados e em percentagem (pode-se trabalhar com metros ou com pés). Assim por exemplo num edifício de apartamentos, analisando várias soluções para um projecto final podemos escolher por exemplo a que tenha a menor área de circulação (a menos que o cliente goste de grandes passeios dentro de casa) não só por questões de funcionalidade, mas também por economia de áreas.

Este programa poderá também fazer parte de outros que tenham a ver com áreas (superfície vidrada, superfícies de vários materiais, etc.) como por exemplo o cálculo de perdas de calor que um edifício tem através dos vidros e paredes, ou um programa para calcular a acústica de um espaço.

Quem sabe se algum arquitecto ou engenheiro já realizou algum destes programas, seria interessante se escrevesse para o clube a dizer alguma coisa. Pessoalmente estou interessado em troca de informações com arquitectos ou engenheiros que tenham algum "software" de aplicação em arquitectura.

```
   2 REM "tron 5"
   3 REM **                  **
   4 REM **Santiago Ribas**
   5 REM **                  **
   6 FOR z=1 TO 30: PLOT 8*z,0:
DRAW 0,166: BEEP .09,z: NEXT z
   7 FOR z=1 TO 20: PLOT 0,8*z:
DRAW 250,0: BEEP .09,z: NEXT z
   8 PRINT AT 14,12; FLASH 1;"TR
ON 5"
   9 FOR x=1 TO 30: INK 7: PAPER
 0: BORDER 0: POKE 22527+RND*704
,RND*127: BEEP .09,RND*x: NEXT x
  10 CLS : PRINT AT 0,1;"DESEJA"
;AT 0.16;"1)COMECAR";,,"2)DESCRI
CAO": PAUSE 4e4: IF INKEY$="1" T
HEN CLS : GO TO 20
  15 PRINT "Este programa calcul
a ,soma e  apresenta ordenadame
nte as areas das varias funcoes
de um edificio ,partindo do prin
cipio que os espacos com formas
mais complexas podem-se subdivid
ir num somatorio de triangulos,q
uadrados e rectangulos.": PRINT
AT 20,0;"carregue no c para ver
um exemplo"
  18 PAUSE 4e4: IF INKEY$="c" TH
EN CLS : GO SUB 610
  20 PRINT FLASH 1;"QUANTAS FUNC
OES ?";: INPUT i: PRINT i: PAUSE
 40: CLS
  25 PRINT "carregue no ";: PRIN
T FLASH 1;"M";: PRINT " se vai t
rabalhar  em metros"
  30 PRINT ,, FLASH 1;"FT";: PRI
NT " se vai trabalhar em feet(pe
s)";: INPUT c$
  40 DIM a(i): DIM y$(i,11)
  50 FOR w=1 TO i: CLS
  60 PRINT "Qual a funcao";w;"?"
;: INPUT y$(w): PRINT ;"-->";y$(
w)
  70 FOR x=1 TO 10: BEEP .03,RND
*(x*3): BEEP .04,RND*(x*4): NEXT
 x
  80 LET total=0
  90 PRINT AT 5,0;"quantas areas
 para a funcao        ";y$(w);"-->"
;: INPUT b: PRINT b
 100 FOR n=1 TO b: PRINT AT 10,0
;"Qual o formato da area ";n;" d
a    funcao ";w;"->";y$(w);"?"
 110 PRINT ,,,, FLASH 1;"T";" TR
IANGULO  R RECTANGULO": INPUT A$
: CLS
 130 IF a$="" THEN GO TO 120
 140 IF a$="t" THEN GO TO 290
 150 IF a$="r" THEN GO TO 190
 190 REM **calculo do rectangulo
**
 195 PLOT 96,96: DRAW 60,0: DRAW
 0,-60: DRAW -60,0: DRAW 0,60: P
RINT AT 13,11;"a": PRINT AT 9,15
;"b"
 200 PRINT AT 0,0;"Quais as dime
nsoes da area ";n;"?": INPUT c:
BEEP .09,12: PRINT AT 8,4;c: INP
UT l: BEEP .09,12: PRINT AT 10,4
;l
 205 PAUSE 20
 210 LET x=c*l: BEEP .09,10
 220 IF c$="m" THEN CLS : PRINT
AT 3,0;"area ";n;"->";x;" m2"
 230 IF c$="ft" THEN CLS : PRINT
 AT 3,0;"area ";n;"->";x;" ft2"
 250 LET total=total+x
 260 IF n=b AND c$="m" THEN PRIN
T "total da funcao ";y$(w),total
;"m2 ": PAUSE 300
 270 IF n=b AND c$="ft" THEN PRI
NT "total da funcao ";y$(w),tota
l;"ft2": PAUSE 300
 274 IF n=b THEN LET a(w)=total:
 NEXT w
 275 NEXT n: CLS
 280 GO TO 400
 290 REM **calculo do triangulo*
*
 295 PLOT 96,96: DRAW 30,-60: DR
AW -60,0: DRAW 30,60: PRINT AT 1
2,12;"h": PRINT AT 18,12;"b"
 296 PLOT 96,160: DRAW 0,-42: DR
AW 63,0: DRAW -63,42: PRINT AT 4
,11;"h": PRINT AT 7,15;"b"
 300 PRINT AT 0,0;"Qual a base x
 altura da area ";n;"?";: INPUT
h: BEEP .09,12: PRINT AT 18,10;h
: PRINT AT 8,15;h: INPUT d: BEEP
 .09,12: PRINT AT 4,8;d: PRINT A
T 12,11;d
 305 PAUSE 20
 310 LET x=(h*d)/2: BEEP .09,10
 320 IF c$="m" THEN CLS : PRINT
AT 3,0;" area ";n;"->";x;" m2"
 330 IF c$="ft" THEN CLS : PRINT
 AT 3,0;" area ";n;"->";x;" ft2"
 360 LET total=total+x
 370 IF n=b AND c$="m" THEN PRIN
T "Total da funcao ";y$(w),total
;" m2": PAUSE 300: LET a(w)=tota
l: NEXT w
 375 IF n=b AND c$="ft" THEN PRI
NT "Total da funcao ";y$(w),tota
l;" ft2": PAUSE 200: LET a(w)=to
tal: NEXT w
 380 NEXT n: CLS
 400 REM **ordenacao das areas**
 405 FOR j=1 TO i-1
 410 FOR k=j+1 TO i
 420 IF a(j)>=a(k) THEN GO TO 50
0
 430 LET u=a(j): LET u$=y$(j): L
ET a(j)=a(k)
 440 LET y$(j)=y$(k): LET a(k)=u
: LET y$(k)=u$
 500 NEXT k: NEXT j
 512 LET f=0: FOR w=1 TO i: LET
f=f+a(w): NEXT w
 515 REM **apresentacao dos resu
ltados**
 520 PRINT AT 0,0; PAPER 1;"FUNC
AO":AT 0.13;"AREA";AT 0.20;" %
```

```
 530 FOR w=1 TO i: LET g=(a(w)*1
00)/f
 540 PRINT AT 1+w,0;y$(w);TAB 13
;a(w);TAB 20;g: NEXT w
 560 PRINT AT 17,0;"Para saber o
 preco da area construida entre
com o preco por m2 em _contos_!"
: INPUT s: CLS
 570 LET d=s*f: PRINT "A sua are
a construida custara     ";d;" con
tos."
 580 PRINT AT 20,0;"c para comec
ar": PAUSE 4e4: IF INKEY$="c" TH
EN CLS : GO TO 1
 600 REM **planta paredes**
 610 PLOT 112,120: DRAW 0,-60: D
RAW 2,0: DRAW 0,48: DRAW -2,0: D
RAW 0,12: DRAW -50,0: DRAW -7,-7
: DRAW -2,2: DRAW 7,7: DRAW 54,0
: DRAW 0,-6: DRAW -2,0
 620 PLOT 47,107: DRAW -16,-16:
DRAW 0,-33: DRAW 2,0: DRAW 0,32:
 DRAW 15,15: DRAW 8,-8: DRAW 1,1
: DRAW -9,9
 630 PLOT 32,58: DRAW 24,-24: DR
AW 20,0: DRAW 0,64: DRAW -11,0:
DRAW 0,-1: DRAW 10,0: DRAW 0,-20
: DRAW -10,0: DRAW 0,-1: DRAW 10
,0: DRAW 0,-42: DRAW 13,0: DRAW
25,25
 640 PLOT 112,97: DRAW -28,0: DR
AW 0,1: DRAW 5,0: DRAW 0,14: DRA
W 1,0: DRAW 0,-14: DRAW 22,0
 660 PLOT 89,112: DRAW 7,7: DRAW
 -7,0
 670 PLOT 76,97: DRAW 0,-7: DRAW
 7,7
 680 PLOT 64,97: DRAW 0,-7: DRAW
 -7,7
 690 PLOT 53,113: DRAW 0,-9: DRA
W -5,5: BEEP .09,20
 700 PRINT AT 12,11;"4";AT 11,8;
"3";AT 12,6;"2";AT 8,8;"1";AT 8,
12;"5"
 710 PRINT AT 6,18;"1 entrada";A
T 7,18;"2 estar /comer";AT 8,18;
"3 cozinha";AT 9,18;"4 quarto";A
T 10,18;"5 inst.sanit."
 720 PRINT AT 0,0;"Exemplo de um
a habitacao          unifamiliar (
T1)";AT 20,0;"Carregue no c para
 ver a divisao das areas das var
ias funcoes": PAUSE 4e4: IF INKE
Y$="c" THEN BEEP .09,20: GO TO 7
30
 730 CLS
 740 PLOT 112,120: DRAW 0,-62: D
RAW 2,0: DRAW 0,50: DRAW -2,0: D
RAW 0,12: DRAW -50,0: DRAW -7,-7
: DRAW -2,2: DRAW 7,7: DRAW 54,0
: DRAW 0,-6: DRAW -2,0
 750 PLOT 47,107: DRAW -16,-16:
DRAW 0,-33: DRAW 2,0: DRAW 0,32:
 DRAW 15,15: DRAW 8,-8: DRAW 1,1
: DRAW -9,9
 760 PLOT 32,58: DRAW 24,-24: DR
AW 20,0: DRAW 0,64: DRAW -11,0:
DRAW 0,-1: DRAW 10,0: DRAW 0,-20
: DRAW -10,0: DRAW 0,-1: DRAW 10
,0: DRAW 0,-42: DRAW 13,0: DRAW
25,25
 770 PLOT 112,97: DRAW -28,0: DR
AW 0,1: DRAW 5,0: DRAW 0,14: DRA
W 1,0: DRAW 0,-14: DRAW 22,0
 830 PLOT 88,108: DRAW -38,0: DR
AW 11,11: DRAW 0,-11: DRAW -4,0:
 DRAW 0,-9: DRAW 32,0: DRAW 0,20
 840 PLOT 32,58: DRAW 80,0
 850 PLOT 56,34: DRAW 0,24: PLOT
 88,34: DRAW 0,24: PLOT 33,76: D
RAW 32,0: DRAW 0,24: BEEP .09,20
 860 PLOT 33,90: DRAW 16,0: DRAW
 0,16: DRAW 0,-8: DRAW 8,0: PLOT
 49,90: DRAW 0,-14: BEEP .09,24
 870 PRINT AT 10,8;"a";AT 9,5;"b
";AT 11,5;"c";AT 11,7;"d";AT 13,
6;"e";AT 15,6;"f";AT 16,8;"g";AT
 16,10;"h";AT 15,12;"i";AT 12,11
;"j";AT 8,12;"k";AT 7,9;"l";AT 8
,8;"m"
 880 PRINT AT 0,15;"FUNCAO";AT 2
,15;"ESTAR b,c,d,e,f,g";AT 4,15;
"COZINHAR a";AT 6,15;"DORMIR h,i
,j";AT 8,15;"INST.SAN. k";AT 10,
15;"ENTRADA l,m,...";AT 20,0;"ca
rregue no c para continuar": PAU
SE 4e4: CLS
 890 IF INKEY$="c" THEN PRINT "E
STA HABITACAO DIVIDE-SE NAS SEGU
INTES FUNCOES:",,"estar: 6 areas
,3 triangulos e 3 rectangulos":
PRINT "dormir: 3 areas :1 triang
ulo e 2 rectangulos": PRINT "ins
t.sanitarias :1 area": PRINT "en
trada: 4 areas: 2 rectangulos e
2 triangulos": PRINT "cozinhar:
1 area": PRINT "circulacao ? are
as"
 900 PRINT AT 20,0;"carregue c p
ara comecar": PAUSE 4e4: IF INKE
Y$="c" THEN RETURN
```

Este programa, para o NEWBRAIN, foi publicado no número anterior com "gralhas". O seu autor, PAULO CASTELO, enviou-nos a sua listagem com as necessárias correcções, que publicamos agora. As nossas desculpas.

```
1   REM **  Plot 1  ** REM
5   REM -------------- REM
6   B=1:A=B:REM ### Paulo Castelo ### REM
10 CLOSE#129:OPEN#0,4,"L200":OPEN#129,11
15 GOSUB 500
16 REM "ESTRELA"
20 PLOTRNG(500,250),WIPE,CEN(250,125),DEG
30 FOR I=0TO 1079STEP 8.5
40 PLOTPLA(0,0),TRN(I)
50 PLOTDBY(125,3):NEXT I
60 IF B=1THEN 300
70 REM "DECISAO"
80 IFA=1 THEN A=0:B=0:GOTO 220
85 REM "REDE"
90 FORN=0TO 1
100 FOR I=125TO-124STEP-6
110 PLOT PLA(I,-125),DRW(125,-I,3)
120 NEXT I
130 FOR I=-123TO 125STEP 6
140 PLOT PLA(-I,125),DRW(-125,I,3)
150 NEXT I
160 FOR I=125TO-124STEP-6
170 PLOT PLA(-I,-125),DRW(-125,-I,3)
180 NEXT I
190 FOR I=-123TO 125STEP 6
200 PLOT PLA(I,125),DRW(125,I,3)
210 NEXT I:NEXT N:A=1:B=1
215 REM "VEU"
220 FOR I=125TO-125STEP-2
230 PLOT PLA(I,125),DRW(I,-125,3)
240 NEXT I:FOR I=-124TO 125STEP2
250 PLOT PLA(I,125),DRW(I,-125,3)
260 NEXT I:GOTO 30
290 REM "LEQUES"
300 B=0:FOR N=0TO 1
310 PLOTCEN(375,0)
320 FOR I=180TO 90STEP-2
330 PLOTPLA(0,0),TRN(I),DBY(250,3)
340 NEXT I
350 PLOTCEN(125,247)
360 FOR I=270 TO 362STEP 2
370 PLOTPLA(0,0),TRN(I),DBY(250,3)
380 NEXT I
390 NEXT N:PLOTCEN(250,125):A=1:GOTO 220
500 FOR I=1TO 7:READ A$
510 PRINTTAB(23);A$:NEXT I
520 RETURN
530 DATA ████ █   ███ █████  █
540 DATA █  █ █  █   █  █   ██
550 DATA █  █ █  █   █  █    █
560 DATA ████ █  █   █  █    █
570 DATA █    █  █   █  █    █
580 DATA █    █  █   █  █    █
590 DATA █    ████ ███  █   ███
600 END
```

Programa CUBO 3

Autor: Paulo Castelo
Porto

Este programa demonstra as possibilidades gráficas do computador **NEWBRAIN.**

Os comandos gráficos usam-se introduzindo PLOT, seguido por um ou mais items, separados por vírgulas.

Por exemplo:

PLOT PLACE (16,6), MOVE (16, 46), MOVE (80, 46), MOVE (16, 46).

A instrução PLOT pode conter uma "string" ou um número que será traçada começando na posição definida por PEN.

Por exemplo, o resultado R de um cálculo poderá ser exibido na parte inferior do écran com PLOT RANGE (110,100), PLACE (10,0), "RESULT=", PLACE (38,0), R.

Outra vantagem — a função PEN — permite ao utilizador determinar a posição do traçado gráfico no écran e regular outros items.

Por exemplo:

PRINT PEN(0), PEN(1): REM posição PW=PEN(9).

```
1   REM
2   REM
3   REM       Plot 5
4   REM                  26/2/1983
5   REM
6
7   REM "Paulo Castelo"
8
9
10 OPEN#0,4:CLOSE#6:OPEN#6,6
20 CLOSE#1:OPEN#1,0,"146"
30 CLOSE#11:OPEN#11,11,"#1w229"
40 Plotrange(1000,1200),centre(0,0),Place(0,0)
50 REM Programa retirado de um boletim
55 REM espanhol Para o VIC-20
60 REM (club commodore n.2),feito Por P.
65 REM Masats em 13 de Setembro de 1982.
70  ?"***********************"
80  ?"*                     *"
90  ?"* PROJECCAO DE CUBOS *"
100 ?"*  DE  N  DIMENCOES  *"
110 ?"*                     *"
120 ?"***********************":?
130 INPUT(" No. de Dimencoes : ")n
140 IF n>1THEN 170
150 PUT31
160 ?"F I M":END
170 Plotwipe:?#1:IF n>10THEN 130
180 P=PI/n
190 i=TRUE
200 FOR j=1TO nSTEP 2
210 i=i+1
220 c=i*P
230 x1(j)=COS(c)
240 y1(j)=SIN(c)
250 NEXT j
260 i=n
270 FOR j=2TO nSTEP 2
280 i=i-1
290 c=i*P
300 x1(j)=COS(c)
310 y1(j)=SIN(c)
320 NEXT j
330 f=0
340 FOR j=1TO n
350 f=f+y1(j)
360 NEXT j
370 x0=0
380 FOR j=1TO n
390 IF x1(j)<0THEN x0=x0+x1(j)
400 i1(j)=0
410 NEXT j
420 f=190/f
430 x0=x0-40/f
440 y0=0
450 ox=x0*f:oy=y0*f
460 FOR i=1TO 2^n-1
470 FOR j=1TO n
480 i2(j)=i1(j)
490 NEXT j
500 FOR j=1TO n
510 IF i1(j)=1THEN 710
520 i2(j)=1
530 x=0
540 y=0
550 FOR k=1TO n
560 x=x+i1(k)*x1(k)
570 y=y+i1(k)*y1(k)
580
590 NEXT k
600 GOSUB 810
610 Plotplace(xd,yd)
620 x=0
630 y=0
640 FOR k=1TO n
650 x=x+i2(k)*x1(k)
660 y=y+i2(k)*y1(k)
670 NEXT k
680 GOSUB 810
690 Plotmove(xd,yd)
700 i2(j)=0
710 NEXT j
720 j=1
730 IF i1(j)=0THEN 770
740 i1(j)=0
750 j=j+1
760 GOTO 730
770 i1(j)=1
780 NEXT i:?#1,n
790 GET#6,a:IF a=0THEN 790
820 xd=x*f*9x+ox+oa
800 GOTO 130
810 9x=3:9y=4.7:oa=450:ob=80
820 xd=x*f*9x+ox+oa
830 yd=y*f*9y+oy+ob
840 RETURN
```

Tabela dos valores através dos quais se pode usar a função PEN

0 — Coordenada X
1 — Coordenada Y
2 — Ângulo
3 — Côr
4 — Côr de fundo
5 — Modo
6 — Côr do ponto na posição actual
7 — Endereço da 1.ª posição de memória na exibição de alta resolução
8 — Endereço da última posição de memória na exibição de alta resolução mais um
9 — Extensão da exibição (Pixels)

Um traçado gráfico estabelece-se usando OPEN para abrir a stream 11.

Um gráfico de alta resolução ocupa mais memória do que um caracter vulgar e por isso pode ser necessário abrir uma área superior.

Por exemplo:

```
OPEN # 6,0,1,"170"
OPEN # 1,11, "#6"
```

EXEMPLO — Verificar se o caracter à direita da posição (p,q) é "branco" e se isso se verificar, mover o caracter que está em (p,q) uma posição para a direita.

```
PUT 22,p+1,q:GET C
IF C=32 THEN PUT 8, 20;GET C:
    PUT 32,C;p=p+1
```

O cursor está posicionado à direita de (p,q) e o caracter obtido é (C). Se o caracter é branco (C=32), o cursor é movimentado para a esquerda e obtém-se o caracter dessa posição. É então escrito um espaço, o que faz mover o cursor para a direita seguido pelo caracter C.

O USO DO NEWBRAIN

O Editor de Écran do NEWBRAIN proporciona facilidades para determinar caracteres situados em qualquer ponto do écran, ou colocar esses mesmos caracteres directamente nas posições seleccionadas.

Isto é obtido com os caracteres de CONTROL designados a seguir, que podem ser usados dentro de um programa BASIC em qualquer linha, através do comando PUT.

PUT 22,X,Y — Posiciona o cursor em X,Y
PUT 22,X,Y,C — Coloca o caracter "C" na posição X,Y
PUT 22,X,Y,20:GETC — Determina qual o caracter que está na posição X,Y
PUT 12 — Mover o cursor para a posição 1,1
PUT 11 — Mover o cursor uma linha acima (se possível)
PUT 10 — Mover o cursor uma linha abaixo (se possível)
PUT 8 — Mover o cursor para a esquerda (se possível)
PUT 26 — Mover o cursor para a direita (se possível)
PUT 21:GET X,Y — Determina a posição do cursor

Obter a linha completa onde o cursor é posicionado
PUT 5:LINPUT(" ")a$
(exemplo ao lado)

INTRODUÇÃO À PROGRAMAÇÃO ESTRUTURADA INTRODUÇÃO À PROGRAMAÇÃO ESTRUTURADA

(Contin. do n.º anterior)

FILAS

Uma fila contém certas semelhanças com uma pilha, mas uma pilha "aberta". Efectivamente, uma fila é um grupo de elementos que possibilita a realização das três operações PUSH, POP e EMPTY. Contudo, ao contrário da pilha, uma fila dá saída ao primeiro elemento que foi introduzido; Daí a expressão inglesa FIFO ("First In First Out") — primeiro a entrar, primeiro a sair.

Estas estruturas são utilizadas principalmente em programas de simulação, representando a espera de pessoas ou acontecimentos em filas intermédias de entrada/saída ou, de modo geral, em gestão de processos nos sistemas de exploração multi-tarefas: por exemplo, a espera de programas até se dispôr de uma impressora deve ser feita através de uma fila.

A implantação física de uma fila faz-se geralmente a partir de um quadro e dois ponteiros — um representa a entrada, o outro representa a saída da estrutura (fig. 1).

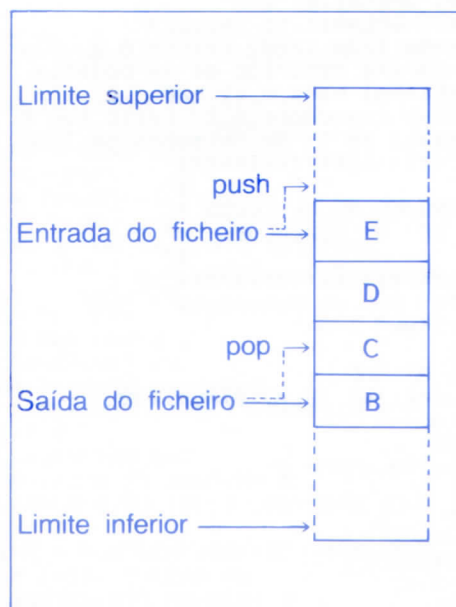



Fig. 1 — A representação física de uma fila utiliza um vector e dois ponteiros: um para indicar a entrada e outro a saída do ficheiro, os quais são incrementados aquando da execução das instruções "colocar" e "retirar".

Em consequência da inserção e da leitura dos elementos pela incrementação dos ponteiros, há uma deslocação contínua da pilha para a parte superior do quadro. Assim, quando um ponteiro chega aos topo passa a zero de modo a apontar para a base do quadro e poder continuar a sua tarefa. Quando o ponteiro de entrada atinge o de saída, a fila está cheia. Se, ao contrário, é o ponteiro de saída que atinge o de entrada, a fila está vazia.

Os procedimentos BASIC que permitem gerar uma fila estão indicados na figura 2.

```
 90 DEF FNVIDE(X)=(PS = PE)
100 DIM FILE (100)
110 REM PE ponteiro de entrada
120 REM PS ponteiro de saída
130 REM colocar (X)
140 PE=PE+1
150 IF PE = 101 THEN PE = 1
160 IF PE = PS THEN PRINT "FILA CHEIA"
    :STOP
170 FILE (PE) = X
180 RETURN
190 REM ::::::::::::::::::
200 REM retirar
210 REM resultado em X
220 IF FNVIDE(X) THEN PRINT "FILA VAZIA"
    :STOP
230 X= FILE (PS)
240 PS = PS + 1
250 IF PS =101 THEN PS =1
260 RETURN
```

Fig. 2 — Os procedimentos de gestão de uma fila escrevem-se frequentemente em BASIC.

ESTRUTURAS EM CADEIA

Vamos agora entrar nas estruturas "muito dinâmicas". Alguns autores entendem que as pilhas e as filas têm uma classificação à parte: incluem-nas na categoria das estrutras semi-estáticas, usando o termo estruturas dinâmicas para aquelas que vamos examinar agora.

As estruturas dinâmicas constituem a "vida" da informática, o seu aspecto, mutante e evolutivo. Nenhum sistema lógico seria possível actualmente, se não existissem tais entidades. Fisicamente, como iremos ver, o elemento essencial é o ponteiro. Ao contrário dos quadros, em que os elementos estão sabiamente alinhados uns ao lado dos outros, os componentes das estruturas dinâmicas estão dispersos no espaço disponível de memória, e ligados a elas graças aos ponteiros.

Apontar significa fazer referência a um elemento sem o nomear explicitamente.

Certas linguagens falam de ponteiros (PASCAL, C), outras falam de referências (ALGOL, SIMULA), e outras de acessos (ADA). Muitas linguagens que não falam directamente de ponteiros assentam nesta noção (LISP, LOGO, APL) e integram-se nas estruturas de dados muito específicas.

LISTAS LINEARES

Uma lista linear descreve-se logicamente como uma sequência ordenada de dimensões variáveis, constituída por elementos de um tipo determinado, a partir da qual se tornam possíveis certas operações.

Denominamos uma lista por $L = (e_1, e_2, \ldots e_{n-1}, e_n)$. As operações são as seguintes:

- O acesso a um elemento particular da lista não se efectua por intermédio de um índice, mas pela relação com um outro elemento da lista graças às funções: "primeira (L)" que reproduz o primeiro elemento da lista, e "seguinte (L)" que reproduz a lista privada do seu primeiro elemento.
- A inserção de um elemento na lista.
- A supressão de um elemento da lista.
- Testar se a lista está vazia ou não.

Portanto, utilizar-se-ão listas lineares sempre que se tenha um conjunto de elementos de dimensão va-

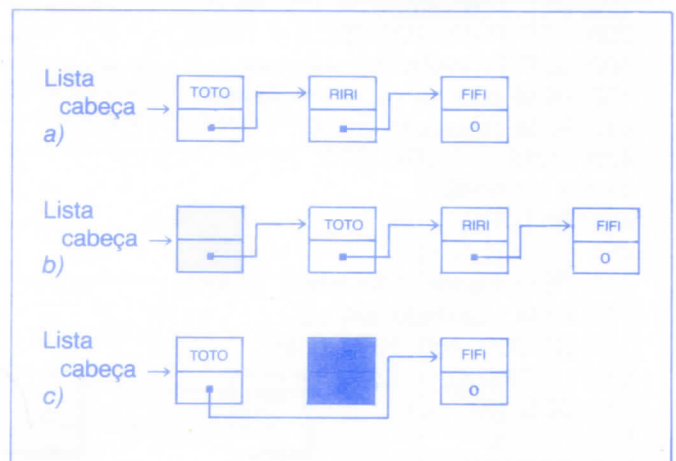



Fig. 3 — Uma lista incluindo 3 elementos (a) pode ver-se prolongada pela inserção de um elemento (b) ou abreviada pela supressão de um deles (c).

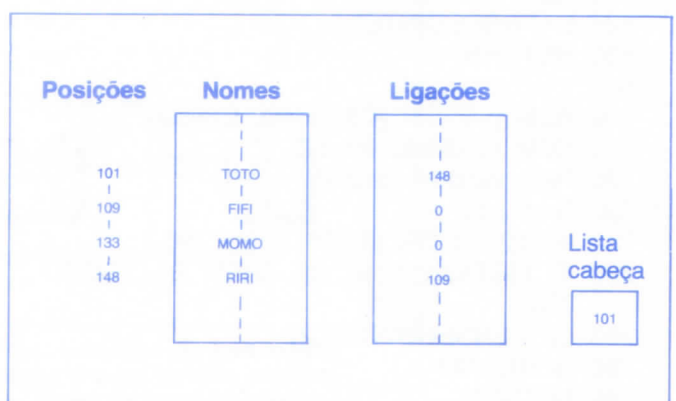



Fig. 4 — Uma lista representa-se na memória em forma de um vector duplo: o primeiro inclui o valor dos elementos, o segundo contém os ponteiros relativos aos elementos sucessivos.

riável (ao contrário dos quadros que são geralmente de dimensão fixa), no qual as operações de inserções, de supressões e de acessos devem ser realizadas.

Um texto no qual se queira inserir ou suprimir linhas, é um bom exemplo de listas lineares. Os elementos são as linhas do texto, e a lista é o próprio texto.

Os sectores num disco são também dispostos em forma de listas lineares, permitindo uma compensação dinâmica dos recursos da memória de massa.

A implantação de uma lista corresponde a uma estrutura encadeada, ou seja, a um conjunto de elementos ligados entre si por ponteiros. A figura 3 mostra esquematicamente as operações de inserção no início da lista e de supressão de um elemento.

As listas podem ser representadas fisicamente sob a forma de um vector duplo: o primeiro contém os elementos, o segundo os ponteiros sobre os elementos, como mostra a figura 4. As rotinas de manipulação para uma tal representação física são dadas em BASIC — figura 5.

```
 90 DIM LISTA$(100),SEG(100)
100 DEF FNPRIM(L) = LISTA$ (L)
200 DEF FNSEGUINTE(L) = SEG(L)
300 DEF FNVAZIO(L) =(L =0)
400 REM criar um novo elemento
410 REM resultado em X
420 LIVRE =LIVRE + 1
430 X =LIVRE
440 RETURN
450 ......................
500 REM inserir (C$:cadeia, L1:lista)
510 REM resultado em L2
520 GOSUB 400: REM criar
530 LISTA$(X) = C$
540 SEG (X) =L1
550 L2 =X
560 RETURN
570 ......................
600 REM suprimir (L: lista)
610 REM resultado em L
620 L=FNSEGUINTE(L)
630 RETURN
640 ......................
700 REM procurar (C$:cadeia, L1:lista)
710 REM resultado em L2
720 Rem variável local B
730 B=1 : L2 = L1
740 IF (L2=0) OR (B=O) THEN 790
750 IF LISTA$(L2) = C$ THEN B=O:GOTO
    780
760 L2 =FNSEGUINTE(L2)
780 GOTO 740
790 RETURN
```

Fig. 5 — As rotinas de manipulação de uma lista linear em BASIC.

Uma outra possibilidade de representação, susceptível de ser utilizada principalmente em linguagens de alto nível (PASCAL, ADA ou C), consiste em exprimir o elemento da lista como um agregado (record) de duas perspectivas: "valor", que contém informação e "seguinte" que é o ponteiro no elemento seguinte da lista. Utilizaremos anotação de PASCAL para descrever este tipo de implementação.

Um ponteiro em PASCAL define-se pela instrução

```
var L: ↑ elemento;
```

que indica que L é um ponteiro sobre o tipo "elemento".

Os ponteiros em PASCAL podem tomar um valor particular nulo, que significa que o ponteiro não referencia nenhum elemento. Para definir uma lista de sequências de caracteres, PASCAL emprega a notação:

```
type lista =↑ elemento;
     elemento=record
              valor:string;
              seguinte: ↑ elemento;
            end;
```

"lista" é um tipo definido como um ponteiro dirigido a elementos; "elemento" como um agregado constituído de uma sequência de caracteres e de um ponteiro no elemento seguinte da lista.

As rotinas de utilização de uma lista linear são dadas em PASCAL, figura 6. O acesso a um valor de um elemento é efectuado pela instrução:

```
P↑ . valor
```

que representa a perspectiva "valor" do elemento apontando por P.

Utilizamos na nossa definição o tipo "string", que não existe nos tipos pré-definidos do modelo ISO, mas que se encontra em inúmeros PASCAL disponíveis em micro-computadores como o U.C.S.D. PASCAL.

A estrutura de lista linear pode ser melhorada de duas maneiras diferentes:

1. Pelo emprego de um encadeamento duplo, a fim de obter uma lista linear dupla (Fig. 7-a)

2. Pela reciclagem do fim da lista no primeiro elemento, para constituir uma lista circular (Fig. 7-b).

Estas novas estruturas permitem remediar certos inconvenientes da estrutura linear simples: leitura dos elementos nos dois sentidos (listas lineares duplas), diminuição da importância atribuída ao primeiro elemento da lista (listas circulares).

```
programa gestaolista;
  type lista = ↑ elemento
    elemento = record
                 valor: car;
                 seg : lista;
               end;
  função primeira (a:lista):car;
    begin
      primeira:=a↑.valor;
    end;
  função seguinte (a:lista):lista;
    begin
      seguinte:=a↑.seg;
      end;
  função vazia (a:lista):booleano;
    begin
      if a=nil then vazia:=true
                else vazia:=false;
    end;
função inserir (c:car;a:lista):lista;
  var p:lista;
  begin
    new(p); (* criação de um novo
                  elemento *)
    p↑.valor := c;
    p↑.seg :=a;
    inserir := p;
end;
função procurar (c:car;a:lista):lista;
  (* reproduzir a lista cujo primeiro
    elemento começa por c *)
  var b:booleano;
  begin
    b:=true;
    while (a< >nil) and b do
      if c = a.valor
        then b:=false
        else a:=seguinte(a);
    procurar:=a;
  end;
begin
end.
```

Fig. 6 — Em PASCAL, a representação de uma lista efectua-se directamente através de ponteiros e de agregados (record). A escrita das rotinas de manipulação de lista é também simplificada.

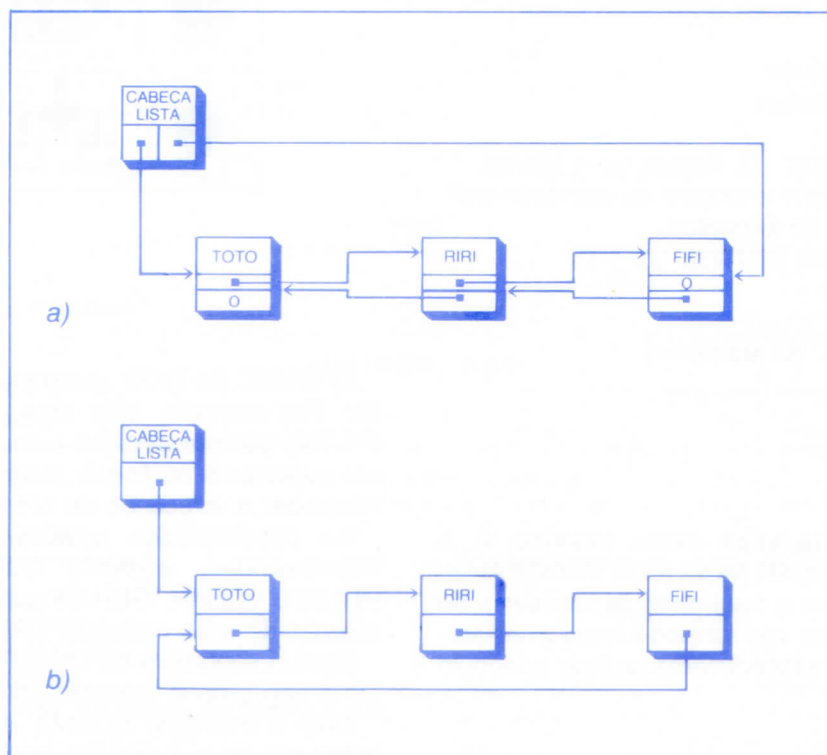



Fig. 7 — São possíveis outras estruturas de listas. As listas lineares duplas (a) permitem uma manipulação nos dois sentidos, enquanto que as listas circulares (b) tornam "iguais" os elementos suprimindo o caracter particular das cabeças de lista.

(continua nos próximos números)

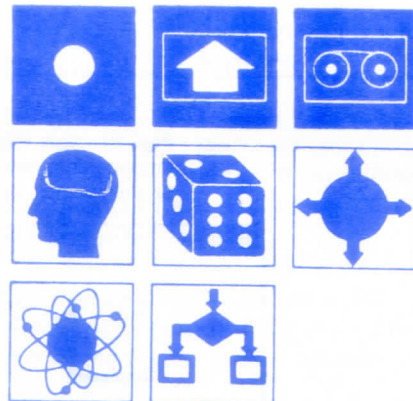
CARTÃO DE APRESENTAÇÃO

Nome: TEXAS TI 99/4A
Fabricante: TEXAS INSTRUMENTS
Importador em Portugal: TEXAS INSTRUMENTS
Preço provável: Esc. 35 000$00/Unidade base
Data de Comercialização: Março/Abril 1983

Home Computer Texas Instruments TI 99/4A

CARACTERÍSTICAS PRINCIPAIS

Microprocessador: TEXAS 9901 — 16 bits
Memória ROM: 26 Kbytes (14 Kbytes p/ Basic)
Memória RAM: 16 Kbytes
Teclado: Qwerty c/ 48 teclas, das quais 15 programáveis
Resolução Video: 192x256 pontos; 20 linhas de 40 caracteres
Dimensões (unidade base): 259x381x71 mm
Peso: 2,3 Kg.
Saídas: Leitor de cassetes
Leitor de diskettes
RS 232 C
Extensões de memória: 32 Kbytes ou 4 Kbytes
Monitor Video: TV preto e branco ou cor com UHF
Periféricos: Unidade de Diskettes
Impressora (RS 232 C)
Gravador
Linguagens: BASIC
PASCAL (c/ adicional)
LOGO

A **principal característica** desta máquina é a POSSIBILIDADE DE USAR MÓDULOS PROGRAMADOS, que não possuem a fragilidade das cassetes ou diskettes e que entram em funcionamento imediatamente, com o simples procedimento de ligar o módulo à consola.

O BASIC do TI/99 apresenta uma certa originalidade: Por exemplo, com uma nova tecla FUNCTION (FCTN), permite associar outras teclas e assim: suprimir caracteres ou linhas; inserir novos elementos; interromper a execução de um programa, etc.

As possibilidades gráficas e sonoras são bem desenvolvidas, existindo SUBPROGRAMAS como HCHAR, VCHAR, GCHAR que podem ser chamados, através de uma instrução CALL.

Com o subprograma CCHAR o utilizador pode definir o seu próprio caracter gráfico.

Com a instrução SOUND, as frequências musicais estendem-se a 4 oitavas, podendo ainda ser variada a amplitude do som.

SOLID STATE SPEECH SYNTHESIZER — reproduz electronicamente a voz humana — requer outros acessórios como o SPEECH EDITOR, TERMINAL EMULATOR II, . . .

CENTRO DE MICROCOMPUTADORES

AMADEU PRAÇA & FONSECA, LDA.
RUA DE SANTA CATARINA, 715 - Loja 0
4000 PORTO
TELEF. 314506

PREÇÁRIO

SINCLAIR

Zx81c/ fonte de alimentação e manual	9 000$00
Módulo de 16k Ram	4 500$00
Módulo de 32k Ram (Memotech)	8 500$00
Módulo de 64k Ram (Memotech)	15 500$00
Impressora ZX Printer	9 500$00
Módulo de gráficos de alta resolução	10 000$00
Módulo gerador de som	4 500$00
Teclado em Kit para Zx81	5 300$00
Consola com teclado para Zx81	8 000$00
Interface Centronics p/ impressora Seikosha, Epson ou OKY c/ cabo	10 000$00
ZX Spectrum 16k Ram c/ fonte de alimentação e manual	19 000$00
ZX Spectrum 48k Ram c/ fonte de alimentação e manual	26 000$00

NEWBRAIN

Newbrain modelo AD (c/ visor integrado)	57 500$00
Módulo de 64k Ram (preço previsto)	16 500$00
Módulo de 512k Ram (preço previsto)	97 900$00
Módulo de comunicações série (8 canais) (preço previsto)	31 900$00
Módulo de comunicações série (32 canais) (preço previsto)	100 500$00
Unidade de diskettes 100k (preço previsto)	55 000$00
Unidade de diskettes 1M (preço previsto)	99 000$00

BBC

BBC modelo A (16k Ram)	70 200$00
BBC modelo B (32k Ram, c/ interfaces)	90 000$00

JÚPITER ACE

Júpiter ACE (linguagem FORTH)	16 500$00

TEXAS INSTRUMENTS

TI-99/4A (preço previsto)	35 000$00

VIC 20

Commodore VIC c/ fonte de alimentação e manual	34 000$00
Unidade de cassetes para VIC 20	9 500$00
Módulo Super Expander (3k Ram e comandos de gráficos e som)	6 840$00
Módulo Programmers Aid	6 840$00

COMMODORE 64

Commodore 64k	75 000$00
Drive	72 000$00

ATARI

Atari 400 16k, c/ Basic, fonte de alimentação e manuais	52 800$00
Atari 800 16k, c/ Basic, fonte de alimentação e manuais	84 500$00
Gravador Atari	10 600$00
Unidades de diskettes Atari	63 200$00

ORIC

Oric c/ 16k Ram	21 000$00
Oric c/ 48k Ram	35 500$00
Módulo de comunicações	16 800$00

IMPRESSORAS SEIKOSHA

Seikosha GP80M (papel de 8 pol., 30 cps)	34 400$00
Seikosha GP100A Mark II (papel de 10 pol., 50 cps)	44 000$00
Seikosha GP100A Mark II NB (p/ Newbrain, papel de 10 pol., 50 cps)	48 000$00
Seikosha GP100V (p/ VIC 20, papel de 10 pol., 30 cps)	40 150$00
Seikosha GP100DB (p/ Sharp, papel de 10 pol., 30 cps)	55 450$00
Seikosha GP250X (papel de 10 pol., 50 cps)	51 000$00
Seikosha GP250X (p/ Newbrain, papel de 10 pol., 50 cps) c/ cabo	52 870$00

IMPRESSORAS EPSON

Epson MX80 FT/3 (papel de 10 pol., 80 cps)	76 500$00
Epson MX80 FT/3 (para Newbrain, papel de 10 pol., 80 cps)	86 000$00
Epson MX100 FT/3 (papel de 14 pol., 100 cps)	97 300$00
Epson MX100 FT/3 NB (para Newbrain, papel de 14 pol., 100 cps)	106 800$00

IMPRESSORAS OKI

Oki Microline 82A (gráficos, papel de 10 pol., 120 cps)	78 000$00
Oki Microline 83A (gráficos, papel de 14 pol., 120 cps)	110 000$00

MONITORES DE VIDEO

Monitor Hitachi 12 polegadas, fósforo verde	23 000$00
Monitor Zenith 12 polegadas, fósforo verde	17 000$00

COMPUTADORES PARA USO PROFISSIONAL

APPLE IIe — package 1
Unidade base c/ 64K de memória
Drive p/ diskette 140K; c/ controlador
Monitor de fósforo verde — Esc. 249 978$00

APPLE IIe — package 4
Mesmo conjunto que o package 1, acrescido de
Drive adicional p/ diskette de 140K
Interface p/ impressora
Impressora Apple — Esc. 420 649$00

PEÇA INFORMAÇÕES MAIS PORMENORIZADAS
SOBRE TODOS OS CONJUNTOS POSSÍVEIS DO EQUIPAMENTO APPLE

Computador SIRIUS,
c/ 128 K de memória e 2 Drives p/ diskettes (600 K) cada — Esc. 579 000$00

Computador SIRIUS
c/ disco de 10 Mbytes — Esc. 1 080 000$00

PEÇA UMA DEMONSTRAÇÃO DOS PROGRAMAS P/ CONTABILIDADE
APPLE / SIRIUS

SOFTWARE para o microcomputador SPECTRUM

Jogos I	1 000$00	Ficheiro Biblioteca	600$00
Xadrez	600$00	Índice Bibliográfico	600$00
Sim. Voo	600$00	Cálculo de Pórticos	2 000$00
Estatística	600$00	Análise de Investimentos	1 000$00
Mat. I Res. Eq.-Matrizes	600$00	Vu File	1 000$00
Jogos Div. (cada)	500$00 ou 600$00	Vu Calc	1 000$00

SOFTWARE para o microcomputador ZX 81

Aplicações profissionais (cada) 1 000$00
Contas Correntes — Salários — Stocks — Análise de Vendas
Análise de Investimentos — Resumos de Facturas, etc.
Jogos diversos (cada) 500$00

SOFTWARE para o microcomputador TEXAS TI/99

Módulos Educacionais	(cada)	3 250$00
Jogos diversos	(cada)	3 250$00
PARSEC	(cada)	5 350$00

SOFTWARE para o Microcomputador New Brain

Engenharia Civil (Cálculo de Estruturas)	15 000$00
Stocks	2 000$00
File (Ficheiros)	1 500$00
Análise de Investimentos	1 500$00
Estatística	1 500$00

SOFTWARE PARA APPLE E SIRIUS

Peça uma lista completa do Software disponível

Vamos organizar brevemente, sessões de apresentação e utilização dos computadores APPLE II — APLE III — SIRIUS, incluindo o aproveitamento do equipamento, como auxiliar de gestão.

Essas sessões, serão limitadas em número de participantes. Se está interessado em inscrever-se para essas jornadas (duração de um dia); escreve-nos, indicando: Nome, ocupação, firma e problemas de gestão com mais interesse na sua actividade profissional.